REMARKS

This communication is responsive to the Office Action dated January 30, 2006. In the Office Action, the Examiner continues to apply the Cyran reference to the claims, but now the Li reference is combined with the Cyran reference. In addition, an obviousness-type double patent rejection has been made.

Claim Amendments

Applicant has further amended the independent claims to clarify that the "JAVA runtime environment features" recited in the "wherein" clause at the end of each independent claim, as being controlled, refer to the "JAVA objects of interest of the runtime environment, for the particular JAVA application." In addition, the claims have been amended, where appropriate, so that the use of "JAVA" as a trademark is proper.

Prior-Art Based Rejection

The claims are rejected as being obvious over the combination of Cyran and Li. Applicant respectfully submits that the combination does not yield what is claimed and that, in any event, the "suggestions" posited by the Examiner to combine Cyran and Li further fall short of what is claimed.

In particular, by contrast to controlling how "objects" of the JAVA runtime environment are treated during execution of a JAVA application, as interpreted by the Examiner, Cyran is directed to controlling (optimizing?) the operation of a JIT compiler. That is, Cyran discloses preprocessing bytecodes of a JAVA application and providing information of that preprocessing to the JIT compiler. Cyran does not disclose generating any information (let alone optional attributes that are written into an attribute table portion of a JAVA class file) that can be or is used by a JAVA virtual machine to control how "objects" are treated during execution of the JAVA application.

In addition, claims 26, 28 and 30 have been amended to recite that "the optional attributes indicate to the JAVA virtual machine which objects of interest of the Java runtime environment need to be loaded for the particular JAVA application." Certainly, Cyran's preprocessing for providing information to a JIT compiler does not indicate which objects of interest need to be loaded for the particular JAVA application. At best, the preprocessing informs the JIT compiler whether the preprocessing for optimization has been performed and, if so, gives some information about what are those optimizations.

Neither does the Li reference provide such a disclosure. The Examiner quotes Li's claim 1, a portion of which recites:

> wherein the object type information file and the attribute data in the application file are employed together to create the object instance during execution of the application file.

By contrast, Applicant's independent claims recite in part, "receiving as input information indicating JAVA application bytecodes that are associated with JAVA objects of interest" and as discussed above, processing optional attributes, generated based on the optimization information, how the "objects of interest" are treated during execution of the JAVA application.

Li discloses creating an object instance based at least in part on attribute data (in an application file). Li does not disclose controlling how an object is treated based on attribute data that has been generated based on optimization information indicating JAVA application bytecodes that are associated with JAVA objects of interest." With respect to dependent claims 26. 28 and 30, while Li discloses in some manner customizing how an object is instantiated for particular applications, nothing in Li discloses that attributes indicate to a JAVA virtual machine which objects need to be loaded for the particular JAVA application, only how some of the objects that are loaded should be instantiated.

Finally, the Examiner contends that it would have been obvious to modify Cyran in view of Li. The Examiner contends such a combination would have been obvious "because this would provide a more efficient means to deal with optional (customize) attributes in a JAVA class file in a JAVA runtime environment. The Examiner references Li's col. 1, lines 11-13 and col. 2, lines 20-26. Col. 1, lines 11-13, deal with efficiencies in "object instantiation."

Col. 1, lines 11-13, might be considered suggest modifying Cyran so that attribute data generated by Cyran (assuming Cyran can be considered to generate attribute data) is "encapsulated." Nothing in this passage of Li suggests modifying Cyran so as to change the character of any attribute data that may be generated by Cyran. Col. 2, lines 20-26 also expresses a desire for efficiency, noting

> It is therefore apparent that there is a need for methods by which Java applications and objects may be transmitted between platforms, and by which applications may be developed which take advantage of information common to different instances of a particular Java object class, thereby preserving network bandwidth and programming resources. This is particularly true if the Java programming language is ever to be used for applications of any substantial size and complexity.

At best, this passage of Li might suggest modifying the disclosure of Cyran such that various objects are instantiated for a particular application based on application attributes associated with that particular application. Merely encapsulating the application attributes of Cyran does not yield what is recited in the claims.

Obviousness-Type Double Patenting

Once the claims are in a final, allowable form (as to prior art), Applicant will consider whether a terminal disclaimer is appropriate.

Claim Objections

Applicant has amended the claims such that "JAVA" is capitalized and is accompanied by generic terminology.

CONCLUSION

Applicant believes that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP

/ASH/
Alan S. Hodes
Reg. No. 38,185

P.O. Box 70250
Oakland, CA 94612-0250
650-961-8300